

A Machine Learning Approach to Violin Vibrato Modelling in Audio Performances and a Didactic Application for Mobile Devices

Matteo Lionello

Aalborg University

mlionel16@student.aau.dk

Rafael Ramirez

Pompeu Fabra Universitat

rafael.ramirez@upf.edu

ABSTRACT

We present a machine learning approach to model vibrato in classical music violin audio performances. A set of descriptors have been extracted from the music scores of the performed pieces and used to train a model for classifying notes into vibrato or non-vibrato, as well as for predicting the performed vibrato amplitude and frequency. In addition to score features we have included a feature regarding the fingering used in the performance. The results show that the fingering feature affects consistently the prediction of the vibrato amplitude. Finally, an implementation of the resulting models is proposed as a didactic real-time feedback system to assist violin students in performing pieces using vibrato as an expressive resource.

1. INTRODUCTION

Vibrato is one of the most important expressive resources in string instruments. It consists of a modulation of the fundamental frequency and can be characterised by two components which are the frequency (or rate), and the amplitude (or extent). In the violin it is performed through a rhythmical motion of the left arm and twist while pressing with the fingers one string along the instruments neck, changing the tension on the string and so, changing its length. There are different ways to play vibrato, performing it with different rates and extents and even controlling them dynamically during one single note. The lack of information in the traditional scores regarding how vibrato has to be played, provides the performer with a large degree of decision and control over expressive parameters which dictate the nature of her/his music interpretation. The implicit information missing is generally provided by the performer with deviations from what is reported in the scores [1].

The aim of expressive performance modelling research is to investigate the criteria by which the musicians decide which kind of expressive deviations to perform. The underlying idea is that, to some extent, there are some canons or unspoken rules which lead the expressive deviations performed by musicians.

Our study takes the perspective of music education. While music expression is mostly introduced at advanced levels

of musicianship, vibrato can be considered an exception. Vibrato is not explicitly specified in a score, but it is an important expressive resource for even beginner students.

The current work aims at decoding the underlying decisions that a professional musician takes in performing vibrato from score information. By modelling the professional musician decisions regarding how to distribute vibrato over a music piece, a vibrato model can be used by beginner violin students as a reference to compare their vibrato skills with an expert target vibrato for an arbitrary score. The application goal of the study is to build a mobile application to provide students with a didactic tool to help them to practice their vibrato skills. The application shows the vibrato features of one classical music piece for violin, predicted by the trained model, showing at the same time the features detected by the live performance of the user. It provides a real-time reference for the student's vibrato performance.

2. RELATED WORK

2.1 Expressive Modelling.

De Poli introduces[2] three different levels of abstraction in information-processing models.

- The first is the physical level, including all the information that can be measured during a performance like timing and gestures.
- The second level includes the sequence of symbols describing the music representation like the pitch and the duration occurring along the rigo.
- On the highest level there can be found the expressive information associated to the music.

Mixing the first two points together, we can refer expressiveness to the means used by the performer to reach the composers' intention beside to her/his contribution to enrich the performance. The first aspect is in other terms, to correlate the expressive deviations measured from the performer with the distribution of symbols across a score.

When dealing with modelling ordered sequence of instances, the information brought by the neighbours of each instance, could lead to choose different modelling strategies. A common choice falls on using models trained through time-dependent systems Variable Order Markov Model (VOMM) and Recursive Neural Network (RNN) rather than to use features to describe neighbours instances and their

context (micro, small, phrasal, global). In the case of time-dependent systems, the output of the model is processed depending on the information processed through the model in the previous steps. RNN methods are assumed to be difficult to train on musical phrases because of the difficulty in defining them. Thus, we assume that vibrato, compared to other expressive features such as timing and volume dynamics, is not related with large context properties. For this reason we opted to use non-time-depending models, using features including micro scene information. The information has properties such as interval jumps and duration differences among the previous and the following notes.

Rafael Ramirez et al. [3] apply machine learning techniques to model the expression in music performances. The information used to train the models includes note duration and metrical position, dynamics expectations, duration of the previous and the following notes, and the intervals occurring between the neighbouring notes. This method is also used in [4] to study deviations in jazz guitar pieces, predicting ornamentation using the attributes extracted note by note from scores. This second study compared classification algorithms for ornamentation prediction beside to regression algorithms for duration ratio, onset deviation, and energy ratio. Marco Marchini in [5] introduces a method to study the inter-dependences among musicians in bow quartet. Through inter-voice contextual attributes, his work studied how the deviations occurred differently distinguishing solo and ensemble performance with increasing degree of expressive intention.

An example of time-dependent modelling is Magenta by Google Deep Mind [6]. Magenta proposes an LSTM-based recurrent neural network[7] designed to model polyphonic music with expressive timing and dynamics and also to predict the sequence of notes to be played.

In contrast to the general goals of synthesis or analysis described so far in modelling expressiveness in musical performance, the goal of the project presented aims at providing a didactic tool for one specific feature, which is the vibrato. So far, we can state that time-dependent models are useful to extract information which distribution is not delimited on a specific area of the score. Starting from the hypothesis that vibrato features are determined by a limited micro context and not by global attributes, the procedural paradigm we opted for is similar to the one proposed in the works by Marchini and Ramirez.

2.2 Detection of vibrato features.

Usually researchers (e.g. [8] [9] [10]) consider vibrato a floating pitch in a frequency range between 3 and 8-12 Hz. The amplitude of vibrato ranges from half a semitone up to one semitone and half (from 50 to 150 cents). These values are reasonable when considering bow instruments such as the violin. For other instruments like wind instruments, theremins, vocal singing and for electronic instruments, the vibrato needs to be modelled considering other ranges and thresholds given by the material and the geometric properties of the instrument. Moreover, vibrato's amplitude in bow instruments depends generally on the central frequency on which the vibrato floats.

Standard techniques for detecting the vibrato, usually measure the change of the fundamental frequency of a pitch contour which is calculated by spectral based methods (sinusoidal decomposition, harmonic product spectrum, etc.) or time based ones (autocorrelation or YIN method [11]). Once the pitch contour is obtained, the vibrato features are accessible by analytic analysis, by using peaks-picking algorithm[9] or by instantaneous frequency deviation[10], or Fourier analysis techniques as happens in [8], where the vibrato rate and extent are calculated starting from interpolation of sinusoidal components.

Luwei Yang in his phd thesis [12] proposed a novel method based on the Filter Diagonalisation Method (FDM). The work detects vibrato and it estimate its parameters by using Decision Tree and Bayes Rule applied to the output of FDM computed over short time-frames.

3. DESCRIPTION OF THE PROJECT

The work described in this paper aims at explaining which attributes are significant for vibrato and at the same time, providing a didactic tool for beginner students which helps them to understand where and how vibrato should be played in particular music contexts. From audio recordings of violin performances, a model is trained to predict vibrato from scores. The output of the model is rendered to provide the students with a real time comparison between their performance and the output of the trained model. This project gathers together research topics such as music signal analysis, music representation, machine learning, and multimedia programming.

Fig. 1 shows the organization of the tasks described in this paper. Starting from audio files of professional performances, vibrato parameters were collected for each note played, and at the same, time score descriptors were collected for each note. By using the scores attributes describing the notes as features, a model based on machine learning techniques has been trained to predict the corresponding vibrato parameters performed by the professional musician in the recordings. The predictions over some scores have been rendered through an application for mobile device which analysing the performance of a user, provides a real-time feedback to a student practising vibrato techniques.

3.1 Data acquisition:

The project used audio files provided by the TELMI project. The audios were recorded at a sample rate equal to 48 KHz and encoded with 16 bits, using one pick-up Fishman PRO-V20-OVI mounted on the violin and connected via radio. The audio files are recorded by a professional musician performing two classical pieces for violin, which are *Salut d'Amour* by Edward Elgar and one study-work from Michael Nyman. Each audio was accompanied with the respective electronic score using musicXML [13] protocol.

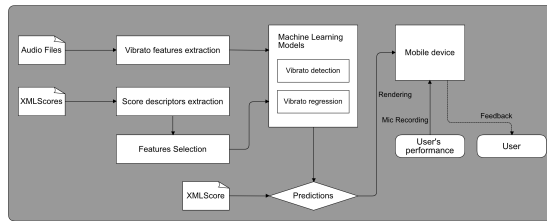


Figure 1. Work flow of the tasks performed

3.2 Vibrato information extraction

We used "Essentia" [14], an open-source library written in C++ for music signals analysis, description and synthesis. Each audio track was automatically processed detecting frequency and amplitude values automatically extracted at a rate of 172.258 Hz. By default, Essentia provides a frequency range of the vibrato estimated from 4 Hz up to 8 Hz, while the amplitude covers a range from 50 cent to 250 cent. The frequency and amplitude are respectively set to 0 Hz and 0 cent if no vibrato was detected or if one of the two values was lower than the minimum of the ranges.

3.3 Pre-processing of the audio data

In order to segment the vibrato values in the corresponding notes, the onset and the duration of each note were extracted by using "Tony" [15]. "Tony" provides an automatic visualisation of audio files by detecting the pitch for each audio frame and rendering the pitch contours. It allows to export the onset and duration of the selected parts. Automatic onset detection on violin audio is not a straight forward procedure because of multiple issues: the energy occurring between two notes linked by a glissando does not necessarily change; when the bow changes its direction during a long note a silent gap is introduced within the same note; the amplitudes is easily controlled by the musician introducing sudden change of energy within a long note; during a glissando it is not obvious to set a threshold where the previous note ends and the next one starts and vibrato notes can be mismatched for multiple notes. Furthermore, the things become even more difficult if, for instance, the interval after a vibrato is one semitone.

The output file consists of 467 records, 242 notes for Elgar's piece and 225 for Nyman's study, including the starting time in seconds and the duration of the windows selected in the program. The starting and the ending of each window have been manually done note by note by removing their attack and the final part of their release (which could be effected by glissandi and other artefacts).

3.4 Vibrato features

The total amount of the records corresponds to the number of the notes written in the scores, including repetitions and rests. The seconds when a note starts and its duration were used to segment the values provided with Essentia according to the notes written in the scores. We decided to use only the mean of all the values computed by Essentia between the start of the note and its offset, given by

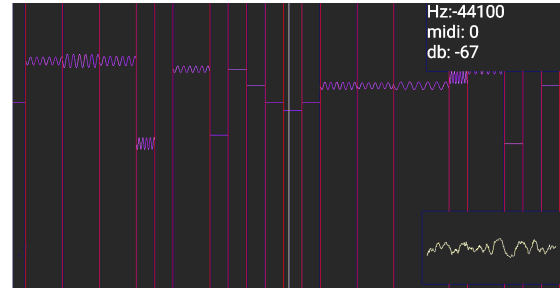


Figure 2. User Interface for the mobile application. The sine waves are modelled according to the vibrato predictions of the notes from a XML score. On the top-right corner the sound level, MIDI note and the pitch are displayed. The actual waveform input is showed in the bottom right corner. The content of the screen scrolls to the left according to the tempo provided in the score

the starting time plus the duration of the note. Other approaches would have considered the note divided in more areas and detecting the modulation of the vibrato throughout all the duration of the note. Once obtained the target values for the model to be computed, the work shifted to the extraction of the features used as input for the model.

3.5 Note descriptors

The XML files have been processed to extract the descriptors note by note. The descriptors consisted in 35 dimensions vectors, describing each note with their pitch, duration, meter position. They included features describing the note's neighbourhood in terms of pitch intervals and beats duration differences respect to neighbour notes. Among the 35 features, there were also included descriptors of perceptual expectation based on Narmour structures.

3.6 Vibrato modelling

There have been computed features selections methods based on information gain and provided with the software Weka [16]. Using the Scikit-learn library[17], a machine learning tools for Python, the data collected was used with 4 different machine learning methods: 1-NN, Decision Tree, ANN, and Support Vector Machine. The 4 methods were used for classification models to detect which notes have to be played with vibrato, and for two regressive models to predict the frequencies and the amplitudes of the notes to be performed with a vibrato. For each different task the model with higher accuracy have been selected.

3.7 Mobile user interface design

Considering the degree of accessibility that the project wants to offer, it has been decided to use a smart-phone context for the rendering section. The general flow chart of the user interaction and the mobile feedback is provided in the Fig. 3. A pilot application has been developed. It shows a first screen on which a list of scores is displayed. By selecting one score the application loads the vibrato predictions already computed for the score and shows a second screen

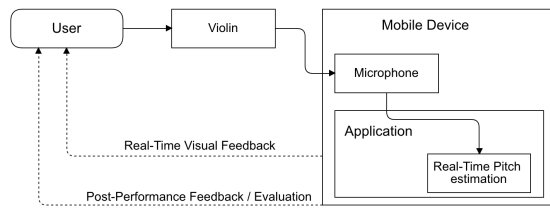


Figure 3. User interaction and device feedback flow chart

displaying the output of the model and the real-time pitch detected from the user's performance, as shown in Fig. 2. During the performance the screen, as shown in 2, displays a sequence of sine waves, each of them representing a note in the score. The sine waves are modelled according to the predictions of the vibrato and placed at the height of screen according to the pitch of the note to which they refer. After 4 beats of a metronome, the sequence of the sine waves describing the notes, start shifting to the left according to the tempo provided with the score. Once the score starts to scroll, the pitch estimated in real time from the microphone is superimposed to the sequence of sine waves. To let the user keep the tempo, a metronome is synchronised with the screen content.

4. FEATURE EXTRACTION AND MODELLING

4.1 Feature extraction

Score processing The score processing saves the information enclosed between the XML tags into a Matlab structure. From each note in the score it extracts information regarding tempo, pitch, measures, accents tides and slurs. The so obtained tables stores onset in beats, duration in beats, midi note pitch, onset and duration in seconds. Since the original code was used for jazz guitar modelling, more attributes were previously considered to include information of chords whose features have been ignored for the current project.

Feature extraction The features are collected in four super-classes. The "Nominal descriptors" refers to the intrinsic properties of the notes such as pitch, tempo, duration, onset and measure. A second class of features includes all the features considering the information provided with the neighbour notes. They include the interval and duration of the previous and next note. "Contextual descriptors" refer to properties inherited by the piece information such as key, note to key and metrical information. The perceptual features are calculated using the midiToolbox library[18] to extract Narmour's IR model's structures (P, D, R, ID, VP, IP) [19] [20]. Tonal stability represents pitch stability respect to the key. Melodic attraction is the weight of the pitches across the pitch space. Tessitura is the standard deviation of the pitch height distribution and predicts the listener's expectation of the tones being close to the median pitch. Mobility is calculated on the expectation of the melody to change direction after long intervals. The features obtained for each note, give a representation of micro, neighbourhood, contextual and perceptual level of the score. Groups of tied notes are considered as a single

note. Their durations are summed and the remainder of the extracted features refers to the first note of the group.

4.2 Learning the Predictive Models

Three models were obtained for three different tasks:

- A vibrato/non-vibrato classification model
- Vibrato amplitude regression model
- Vibrato frequency regression model

For the vibrato/non-vibrato model, a binary classifier was trained, while for the other two tasks we trained regressive models. Each time the first model predicted a note to be performed with vibrato, the other two models were used to predict the vibrato amplitude and frequency. Out of the 467 notes instances in the data set, 200 notes were detected by Essentia as performed with vibrato. The vibrato notes have frequencies mean equals to 5.63 +/- 0.48 Hz ranging from 4.07 Hz up to 6.80 Hz, and extents mean equals to 71.64 +/- 17.10 cent ranging from 50.15 up to 130.99 cent as shown in Fig. 4. Figure 5 shows the comparison between training error and cross validation while increasing the size of the training set, using a ratio between training set size and test set size ranging from 1:1 till approximately 3:1. It displays the results of a Support Vector Machine for the regression tasks and a 1-Nearest Neighbour for the classification task. The progressions show the accuracy obtained for the vibrato detection and the mean square error for the frequency and amplitude prediction. The cross validation has been performed using 10 folds over a shuffling size-fixed test set and a shuffling size-increasing training set. This is a helpful tool to detect problems affecting the training stage. In particular, we can observe that for the frequency regression the distance between the two curves decrease consistently while keeps quite large for the the other two models. All the models were implemented with four different algorithms to compare the results: Decision Tree, Artificial Neural Network, Support Vector Machine and 1-Nearest Neighbour.

4.3 Features selection

Following what previously described in 4.1, the features extracted for each note are the following: duration, pitch, duration of the previous and following note, harmonic interval between the current note and the previous and the

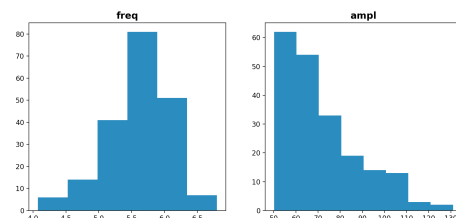


Figure 4. The vibrato notes distributions. Frequencies mean 5.63 +/- 0.48, max: 6.80, min: 4.07. Amplitudes mean: 71.64 +/- 17.10, max: 130.99, min: 50.15

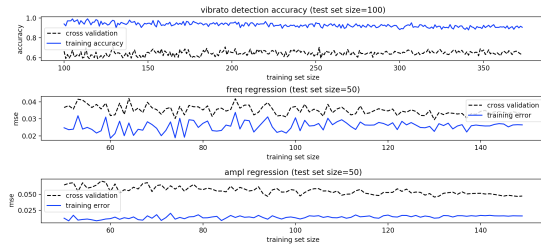


Figure 5. Models' quality comparison between training error and cross validation, increasing the size of the training set.

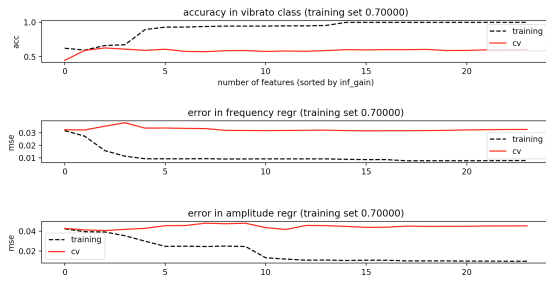


Figure 6. Models' training and 10-cross fold validation augmenting the amount of feature considered during the training, sorted according to their information gain

following, pitch, metre information, pitch to dominant information, registral direction [21], the registral return [21], intervallic difference, closure, the proximity [21] and the consonance [22], the tone stability [23], melodic attraction [24], the tessitura and the mobility [25], and Narmour structures calculated for the target note, the previous and next one. In fig. 6 the mean square error and accuracy for the three tasks, calculated by increasing the number of features used to train the models, is presented. It is easy to see that even the training error improves when considering more features while the testing set decrease its quality.

4.4 Correlation-based Feature Subset Selection

Based on [26], Weka [16] provides a subset of features composed by those chosen for their information gain and minimizing the redundancy among them. The subset of features selected are so highly correlated with the classes and carrying a low inter-correlation. For the frequency regression, the following features have been selected: previous note duration, registral direction and the tessitura, while for the amplitude task the features selected are: duration, next note duration, previous interval, interval of the note respect the dominant, the registral direction and Narmour structure of the previous note. The subset of features selected for the vibrato classification are: duration, the pitch, the next interval and Narmour structure of the next note.

Adding fingering feature to the model To investigate to what extend fingering influences the performance of vibrato, the fingering used by the professional player was manually transcribed for "Salut d'Amour". The finger-

Models output using information from both the scores:

Task:	SVM	ANN	DT	1-NN
mse for Frequency Regr.:	0.1774	0.1916	0.1802	0.2518
mse for Amplitude Regr.:	0.2044	0.2365	0.2003	0.2666
Accuracy for Vibrato Class.:	60.5485	69.8312	68.9873	73.4177

Table 1. The table summarize the results obtained from the modelling of the features from the two scores excluded the feature regarding the fingering

ing was transcribed from a video recorded during the same recording session used for the vibrato feature extraction audios. The total note played are 248, 102 of them are detected by Essentia as vibrato notes. The fingering information (consisting in values {1,2,3,4} corresponding to the finger used to press the string) was selected by Weka for the feature subset for amplitude regression (duration, next note duration, previous interval, note interval respect the dominant, the closer, the consonance, the stability, and the Narmour structure of the current and the previous note and fingering). This proves that fingering is an important source of information when performing a vibrato. However, it results that fingering feature has not been selected in either the vibrato classification feature subset (duration, pitch, previous note duration, tessitura, metre position, Narmour structure for the current and next note) nor in the frequency regression feature subset (duration, previous note duration, registral direction, registral return and tessitura), being not considered a relevant information for these two tasks.

5. RESULTS

Tables 1 and 2 summarise the errors obtained from different models used. The errors are computed using 10-folds cross validation over the datasets. For the regression and the classification models the mean square error and the accuracy are reported.

The reduction of the size of the datasets by considering only the Elgar's work, seems generally to help the detection of Vibrato, while the regression seems to be negatively affected by the poor amount of instances available respect to the whole dataset. Comparing between the presence and the absence of the fingering feature it can be observed that, considering the reduction of instances used, while the frequency seems to be largely affected by the poor amount of samples, the amplitude predictions which use the fingering feature, increased the error by the 9.4% compared to the 17.59% of error increased for the frequency regression. Indeed, figure 5 shows in fact that the training error and the cross validation error are affected by the lack of a sizeable dataset. Moreover Fig. 6, which excludes the usage of fingering feature, suggests that the choice of reducing the amount of feature is effective, avoiding problems of over-fitting. Figure 7 shows the prediction curves of vibrato amplitudes and frequencies compared to the target values of the note played. The predictions were calculated

Models output introducing fingering information only for "Salut d'Amour":

Task:	SVM	ANN	DT	1-NN
mse for Frequency Regr.:	0.2114	0.2379	0.2086	0.2805
mse for Amplitude Regr.:	0.2362	0.3521	0.2192	0.3048
Accuracy for Vibrato Class.:	70.5645	73.7903	70.5645	72.5806

Table 2. The table summarize the results of the models for *Salut d'Amour* introducing the fingering feature.

by using the model performing the best results from the table above. We can see that the values representing the predictions range over the same extent of the real values.

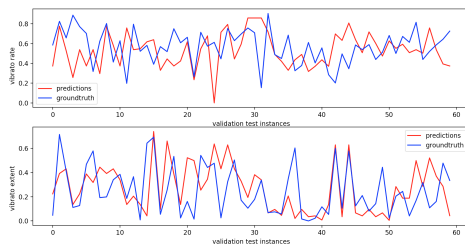


Figure 7. The test predictions and the ground truth values for the two regressive models performing the best results in Table 1.

6. CONCLUSIONS AND FUTURE WORK

This paper provides a study of significant features affecting violin vibrato production. It points out at the importance of fingering information (see section 4.4) in expressive performances. Fingering seems to affect only the amplitude of the vibrato while it seems to not influence its rate. Since fingering is unique for the piece, it provides a low inter-correlated features, letting it to be a potentially suitable attribute for the models. It could be further considered that the piece requires vibrato on those notes which occur to be played mostly with particular fingers.

A real-time feedback application was designed targeting beginner violin students, which allows to compare the students' vibrato choices with those produced by a model trained to take decisions similar to a professional musician.

Future work includes training models with a larger dataset. The results obtained indicate that the accuracy of the models would very likely increase if more data were included in the training process.

Acknowledgments

This work was partly sponsored the European Union Horizon 2020 research and innovation program under grant agreement No. 688269 (TELMi project).

References

- [1] C. Palmer, "Music performance," *Annu. Rev. Psychol.*, vol. 48, pp. 115–138, 1997.
- [2] G. D. Poli, "Methodologies for expressiveness modelling of and for music performance," *Journal of New Music Research*, vol. 33, no. 3, pp. 189–202, 2004.
- [3] R. Ramirez and A. Hazan, "Induction of expressive music performance models," in *Proceedings of the 2004 International Conference on Machine Learning and Applications - ICMLA 2004, 16-18 December 2004, Louisville, KY, USA.*, 2004, pp. 172–177.
- [4] S. Giraldo and R. Ramírez, "A machine learning approach to ornamentation modeling and synthesis in jazz guitar," *Journal of Mathematics and Music*, vol. 10, no. 2, pp. 107–126, 2016.
- [5] R. R. e. a. Marco Marchini, "The sense of ensemble: a machine learning approach to expressive performance modelling in string quartets," *Journal of New Music Research*, vol. 43, no. 3, pp. 303–317, 2014.
- [6] I. Simon and S. Oore, "Performance rnn: Generating music with expressive timing and dynamics," <https://magenta.tensorflow.org/performance-rnn>, 2017.
- [7] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, pp. 1735–80, 12 1997.
- [8] L. Regnier and G. Peeters, "Singing voice detection in music tracks using direct voice vibrato detection," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, April 2009, pp. 1685–1688.
- [9] A. e. a. Friberg, "Cuex: An algorithm for automatic extraction of expwvressive tone parameters in music performance from acoustic signals," *Acta Acustica united with Acustica*, vol. 93, no. 3, pp. 411–420, 2007.
- [10] e. a. Pei-Ching Li, Li Su, "Analysis of expressive musical terms in violin using score-informed and expression-based audio features," in *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, vol. 13, 2015.
- [11] A. de Cheveigné and H. Kawahara, "Yin, a fundamental frequency estimator for speech and music," *The Journal of the Acoustical Society of America*, vol. 111, no. 4, pp. 1917–1930, 2002.
- [12] Y. Luwei, "Computational modelling and analysis of vibrato and portamento in expressive music performance," Ph.D. dissertation, School of Electronic Engineering and Computer Science, Queen Mary University of London, 12 2016.
- [13] M. Good, *Computing in Musicology*, vol. 12, pp. 113–124, 2001.

- [14] D. B. et al., “Essentia: an audio analysis library for music information retrieval,” in *International Society for Music Information Retrieval Conference (ISMIR’13)*, Curitiba, Brazil, 04/11/2013 2013, pp. 493–498.
- [15] M. M. et al., “Computer-aided melody note transcription using the tony software: Accuracy and efficiency,” in *Proceedings of the First International Conference on Technologies for Music Notation and Representation*, May 2015, accepted.
- [16] G. H. B. P. P. R. Mark Hall, Eibe Frank and I. H. Witten, “The weka data mining software: An update,” *SIGKDD Explorations*, vol. 11, 2009.
- [17] F. e. a. Pedregosa, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [18] P. Toiviainen and T. Eerola, “MIDI toolbox 1.1,” <https://github.com/miditoolbox/>, 2016.
- [19] E. Narmour, “The ‘genetic code’ of melody: Cognitive structures generated by the implication-realization model,” 1989.
- [20] —, *The Analysis and Cognition of Melodic Complexity: The Implication-Realization Model*. Chicago: University of Chicago Press., 1992.
- [21] E. G. Schellenberg, “Simplifying the implication-realization model of melodic expectancy,” *Spring*, vol. 14, no. 3, pp. 295–318, 1997.
- [22] C. Krumhansl, “Music psychology and music theory: Problems and prospects,” *Music Theory Spectrum*, vol. 17, no. 1, pp. 53–80, 1995.
- [23] K. E. J. Krumhansl C. L., “Tracing the dynamic changes in perceived tonal organization in a spatial representation of musical keys,” *Psychol. Rev.*, vol. 89, p. 334, 1982.
- [24] L. F., “Calculating tonal tension,” *Music Percept. Interdiscipl. J.*, vol. 13, pp. 319–363, 1996.
- [25] V. H. P., “Redefining pitch proximity: tessitura and mobility as constraints on melodic intervals,” *Music Percept. Interdiscipl. J.*, vol. 17, p. 315327, 2000.
- [26] M. A. Hall, “Correlation-based feature subset selection for machine learning,” Ph.D. dissertation, University of Waikato, Hamilton, New Zealand, 1998.